

LISTING OF AND AMENDMENTS TO CLAIMS:

1. (previously presented) A method for distributing at least one application in a communication network, said method comprising the steps of:

6 /  
redirecting to one server of a plurality of proxy servers at locations distributed throughout the network, at least one service request received from a client for said at least one application;

determining a set of programs required at said one server to fulfil said request for said at least one application; and

executing said set of programs.

2. (original) A method as recited in claim 1, further comprising examining a cache of programs to obtain the set of programs;

3. (original) A method as recited in claim 2, wherein said cache is located at another server of said plurality of proxy servers.

4. (original) A method as recited in claim 2, further comprising returning the results of the step of executing to the client.

5. (original) A method as recited in claim 1, further comprising forwarding a portion of the request that needs to be satisfied at another server to said another server.

6. (original) A method as recited in claim 5, wherein said another server is a backend server.

7. (original) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for distributing at least one application in a communication network, said method comprising the steps of claim 1.

8. (original) A method as recited in claim 1, wherein the step of executing includes:

obtaining parameters for execution from a backend server; and

writing any resulting logging and error messages to said backend server.

9. (original) A method as recited in claim 1, where the step of determining includes parsing the request to determine the program required to satisfy the request.

10. (currently amended) A method as recited in claim 9, further comprising:

retrieving a proxylet-record for said program;  
and

looking up a field of said proxylet-record for determining the set of programs to be executed at the proxy server [[:]].

11. (original) A method as recited in claim 2, where the step of examining includes:

employing a local store in determining a first set of programs present at the first proxy server; and

6  
downloading a second set of programs from another server for said second set of programs not present at said proxy.

12. (original) A method as recited in claim 1, where the step of redirecting is based upon a-priori knowledge of location of said set of programs.

13. (original) A method as recited in claim 12, wherein said a-priori knowledge is deployed at a domain name server.

14. (original) A method as recited in claim 12, wherein said a-priori knowledge is deployed at a backend server.

15. (previously presented) An apparatus to accelerate a distributed application within a network, the apparatus comprising:

a wide area load balancer for distributing at least one request from at least one client to a particular proxy server from among a plurality of proxy servers distributed throughout the network;

an application distributor for distributing a set of programs used for said distributed application to at least said particular proxy server necessary to satisfy said at least one request;

an information-management recorder for recording connectivity information about the set of programs; and

an execution device for executing said set of programs satisfying said at least one request at said particular proxy server.

16. (original) An apparatus as recited in claim 15, further comprising a request forwarder for forwarding to another server any portions of said at least one request which have to be executed at said another server.

17. (currently amended) An apparatus for distributing at least one application in a communication network, comprising:

a plurality of proxy servers each having:

a first set of programs used in said at least one application, a second set of programs retrieved

from a back-end server and executed locally to satisfy part of at least one request received from a client;

a set of cached data associated with said sets of programs;

a set of information-management records for said sets of programs; and

1  
a ~~Cache-Manager~~ cache manager for maintaining the set of programs, the set of cached data and the set of information-management records in distribution of said at least one application.

18. (previously presented) An apparatus for distributing at least one application in a communication network, said apparatus comprising a backend server having:

a first set of programs used for said at least one application, said first set of programs being distributed to a plurality of proxy servers within the network;

a second set of programs used for said at least one application, said second set of programs being executed locally by the backend server;

a third set of programs used for said at least one application, said third set of programs to

receive logging and error messages from the execution of said first set of programs; and

an accessing server to provide access to the first set of programs by any of the proxy servers.

19. (original) An apparatus as described in claim 18, further comprising a request redirector for redirecting requests to one of the plurality of proxy servers.

20. (original) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for causing application distribution in a network with a plurality of machines, the computer readable program code means in said article of manufacture comprising computer readable program code means for causing a computer to effect the steps of claim 1.

21. (previously presented) A method for distributing at least one application, said method comprising:

redirecting one client for said at least one application, to a first proxy server from a plurality of proxy servers distributed throughout a network;

evaluating a request for said at least one application to determine a part of said application that is executable at the first proxy server; and

executing said part at said proxy server.

22. (original) A method as recited in claim 21, further comprising obtaining at least one program used by said at least one application enabling said step of executing.

23. (original) A method as recited in claim 22, further comprising determining a location of said at least one program.

24. (original) A method as recited in claim 22, further comprising obtaining values of parameters specific to said request

25. (original) A method as recited on claim 24, wherein the step of executing includes:

performing at least one operation to satisfy said request; and

writing any resulting logging messages to a backend server.

26. (original) A method as recited in claim 25, wherein said backend server is managing said at least one program.

27. (original) A method as recited in claim 23, wherein said location is the location of a second proxy server.

28. (original) A method as recited in claim 23, wherein the step of determining includes:

obtaining a proxylet-record for said request; and

looking up at least one field in the proxylet-record.

29. (original) A method as recited in claim 24, wherein the step of obtaining includes:

obtaining a proxylet-record for said request; and

looking up at least one field in the proxylet-record.

30. (previously presented) A method as recited in claim 21, further comprising redirecting a second request from said client to a second proxy server.

31. (original) A method as recited in claim 21, further comprising redirecting a second request received from a second client to said first proxy server.

32. (original) A method as recited in claim 21, further comprising redirecting a second request received from a second client to a second proxy server.

33. (original) An article of manufacture comprising a computer usable medium having computer readable program code means embodied therein for causing application



distribution in a network with a plurality of machines, the computer readable program code means in said article of manufacture comprising computer readable program code means for causing a computer to effect the steps of claim 21.

34. (original) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for distributing at least one application in a communication network, said method comprising the steps of claim 21.

35. (previously presented) A method as recited in claim 1, further comprising selecting said one server to be closer to a client requesting a service provided by said application, than other of said plurality of proxy servers.

36. (previously presented) A method as recited in claim 1, further comprising providing administrative control of said application with a backend server, and selecting said one server to be closer to a client requesting a service provided by said application than said backend server.

37. (previously presented) A method as recited in claim 1, further comprising storing at least a portion of said programs on each of said proxy servers.

38. (previously presented) A method as recited in claim 1, further comprising providing a backend server in communication with each of said proxy servers for administrative control of each of said proxy servers.

39. (previously presented) A method as recited in claim 1, further comprising providing a backend server in communication with said client for providing services for portions of said application that are not readily distributable to said proxy servers.

40. (previously presented) An apparatus as recited in claim 15, further comprising means for selecting said particular proxy server to be closer to a client requesting a service provided by said application, than other of said plurality of proxy servers.

41. (previously presented) An apparatus as recited in claim 15, further comprising:

a backend server for providing administrative control of said application, and

means for selecting said particular proxy server to be closer to a client requesting a service provided by said application than said backend server.

42. (previously presented) An apparatus as recited in claim 15, wherein said application distributor distributes at least a portion of said programs to each of said proxy servers.

43. (previously presented) An apparatus as recited in claim 15, further comprising a backend server in communication with each of said proxy servers for administrative control of each of said proxy servers.

44. (previously presented) An apparatus as recited in claim 15, further comprising a backend server in communication with said client for providing services for portions of said application that are not readily distributable to said proxy servers.

45. (previously presented) A method as recited in claim 21, further comprising selecting said first proxy server to be closer to a client requesting a service provided by said application, than other of said plurality of proxy servers.

46. (previously presented) A method as recited in claim 21, further comprising providing administrative control of said application with a backend server, and selecting said first proxy server to be closer to a client requesting a service provided by said application than said backend server.

47. (previously presented) A method as recited in claim 21, further comprising storing at least a portion of said programs on each of said proxy servers.

48. (previously presented) A method as recited in claim 21, further comprising providing a backend server in communication with each of said proxy servers for administrative control of each of said proxy servers.

49. (previously presented) A method as recited in claim 21, further comprising providing a backend server in communication with said client for providing services for portions of said application that are not readily distributable to said proxy servers.

50. (new) A method as recited in claim 10, wherein the proxylet-record comprises at least one of the following:

a requested URL field for the URL that the client was requesting;

an execute URL field for a name of a URL which will be executed at a local proxy;

a code location field for containing a location for code required to run a URL which will be executed at the local proxy;

a parameter list field for any parameters which are passed to the program in order for the program to be executed;

an expiration time field for a date until which the proxylet-record may be considered valid;

a logger URL field for identifying a location where error messages and diagnostic output of the proxylet are provided; and

a code version field for containing the time when the program set identified in the code location field was last modified.

51. (new) A method as recited in claim 50, wherein contents of the parameter list field are different for different proxy servers within the network.

52. (new) A method as recited in claim 50, wherein when a time specified by the expiration time field has expired, a proxy server retrieves a new proxylet-record from a backend server.

53. (new) A method as recited in claim 50, wherein the logger URL identifies a location on a backend server.

54. (new) A method as recited in claim 28, wherein the proxylet-record comprises at least one of the following:

a requested URL field for the URL that the client was requesting;

an execute URL field for a name of a URL which will be executed at a local proxy;

a code location field for containing a location for code required to run a URL which will be executed at the local proxy;

a parameter list field for any parameters which are passed to the program in order for the program to be executed;

an expiration time field for a date until which the proxylet-record may be considered valid;

a logger URL field for identifying a location where error messages and diagnostic output of the proxylet are provided; and

a code version field for containing the time when the program set identified in the code location field was last modified.